Monte Carlo method calculates the probability of a random number

Liu Yin

2020/3/21

Task

Consider approximation of the distribution function of $N(0,1)$,

$$\Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy$$

by the Monte Carlo methods:

$$\hat{\Phi}(t) = \frac{1}{n} \sum_{i=1}^{n} I(X_i \leq t)$$

where X_i's are a random sample from $N(0,1)$, and $I(\cdot)$ is the indicator function. Experiment with the approximation at $n \in \{10^2, 10^3, 10^4\}$ at $t \in \{0.0, 0.67, 0.84, 1.28, 1.65, 2.32, 2.58, 3.09, 3.72\}$ to form a table.

true value for comparison

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
x=pnorm(  c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72), mean = 0, sd = 1, lower.tail = TRUE, log.p =
t=c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72)
n1=10^2
z1=c(rep(0,9))
w1=matrix(0,9,n1)
y1=c(rnorm(n1,mean=0,sd=1))
for(k in 1:9)
{
  for(j in 1:n1)
  {w1[k,j]=sign(y1[j]<=t[k])}
  z1[k]=sum(w1[k,])/n1}

n2=10^3
```

```
z2=c(rep(0,9))
w2=matrix(0,9,n2)
y2=c(rnorm(n2,mean=0,sd=1))
for(k in 1:9)
{
  for(j in 1:n2)
  {w2[k,j]=sign(y2[j]<=t[k])}
  z2[k]=sum(w2[k,])/n2}


n3=10^4
z3=c(rep(0,9))
w3=matrix(0,9,n3)
y3=c(rnorm(n3,mean=0,sd=1))
for(k in 1:9)
{
  for(j in 1:n3)
  {w3[k,j]=sign(y3[j]<=t[k])}
  z3[k]=sum(w3[k,])/n3}


tb<-tibble(
  t=t,
  true=x,
  value1=z1,
  value2=z2,
  value3=z3
)
tb
```

```
## # A tibble: 9 x 5
##       t  true value1 value2 value3
##   <dbl> <dbl>  <dbl>  <dbl>  <dbl>
## 1  0    0.5     0.52  0.512  0.504
## 2  0.67 0.749   0.76  0.727  0.754
## 3  0.84 0.800   0.82  0.779  0.804
## 4  1.28 0.900   0.94  0.88   0.899
## 5  1.65 0.951   0.96  0.947  0.950
## 6  2.32 0.990   0.99  0.987  0.989
## 7  2.58 0.995   1     0.996  0.995
## 8  3.09 0.999   1     0.998  0.999
## 9  3.72 1.00    1     1      1.00
```

repeat the experiment 100 times

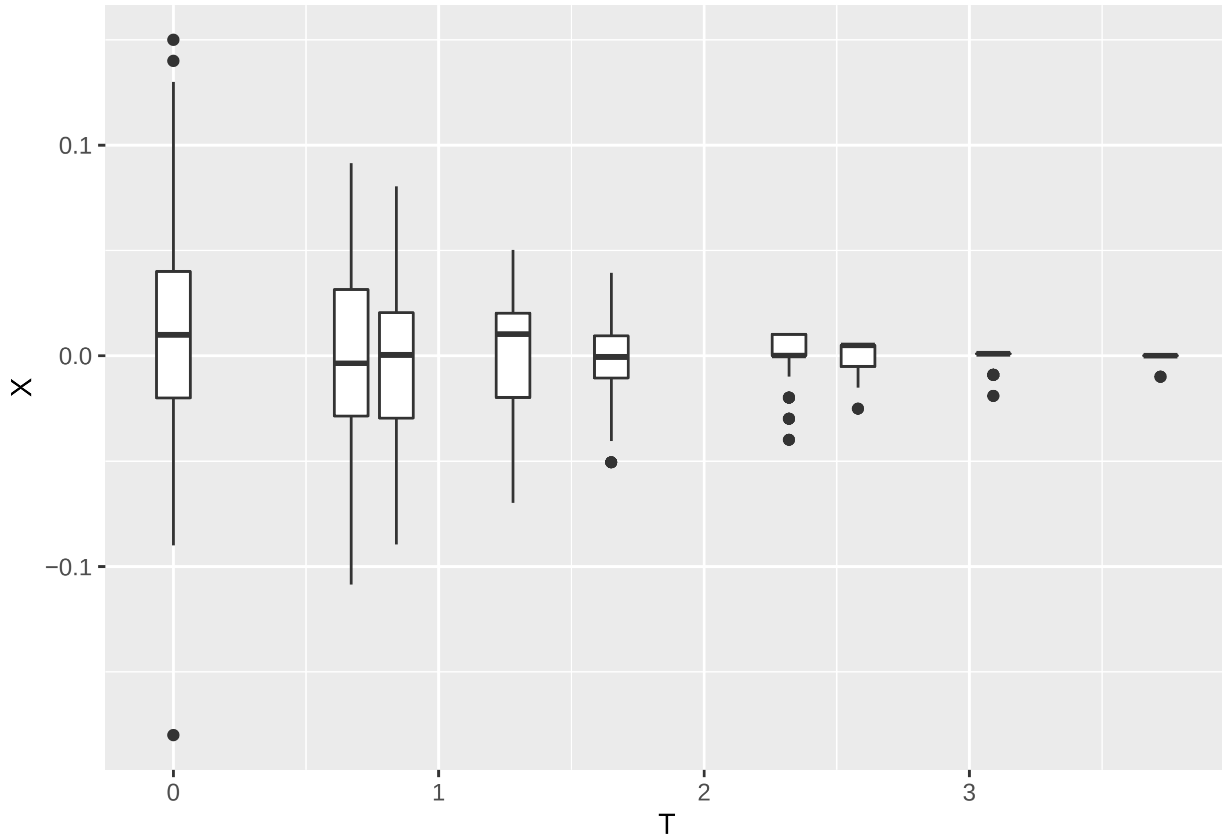
when n=10^2,the box plots of the 100 approximation errors at each t.

```
x=pnorm(  c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72), mean = 0, sd = 1, lower.tail = TRUE, log.p =
t=c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72)

n=10^2
z=matrix(0,100,9)
w=matrix(0,9,n)
```

```r
for(p in 1:100)
{ y=c(rnorm(n,mean=0,sd=1))
  for(k in 1:9)
  {
   for(j in 1:n)
  {w[k,j]=sign(y[j]<=t[k])}
z[p,k]=sum(w[k,])/n}}
z=as.data.frame(z)
r=c(z$V1,z$V2,z$V3,z$V4,z$V5,z$V6,z$V7,z$V8,z$V9)
e=c(rep(0.0,100),rep(0.67,100),rep(0.84,100),rep(1.28,100),rep(1.65,100),rep(2.32,100),rep(2.58,100),rep
q=data.frame(T=rep(0,100),X=0)
for(s in 1:900)
{q[s,2]=r[s]}
for(s in 1:900)
{q[s,1]=e[s]}

for(a in 1:100)
 { q[a,2]=q[a,2]-x[1]
 q[a+100,2]=q[a+100,2]-x[2]
 q[a+200,2]=q[a+200,2]-x[3]
 q[a+300,2]=q[a+300,2]-x[4]
 q[a+400,2]=q[a+400,2]-x[5]
 q[a+500,2]=q[a+500,2]-x[6]
 q[a+600,2]=q[a+600,2]-x[7]
 q[a+700,2]=q[a+700,2]-x[8]
 q[a+800,2]=q[a+800,2]-x[9]}
library(ggplot2)
ggplot(q,aes(T,X,group=T)) + geom_boxplot()
```



when n=10^3,the box plots of the 100 approximation errors at each t

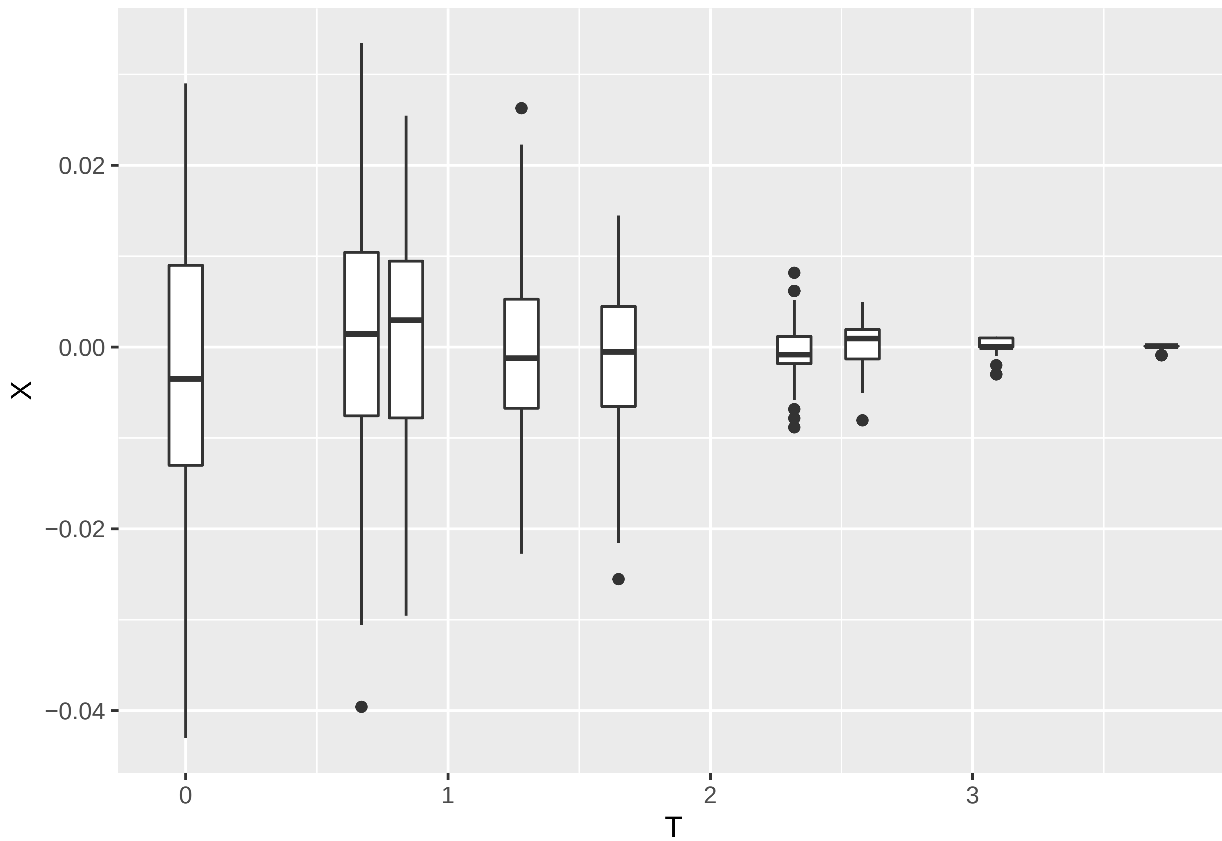
```
x=pnorm(  c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72), mean = 0, sd = 1, lower.tail = TRUE, log.p =
t=c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72)

n=10^3
z=matrix(0,100,9)
w=matrix(0,9,n)

for(p in 1:100)
{ y=c(rnorm(n,mean=0,sd=1))
  for(k in 1:9)
  {
   for(j in 1:n)
  {w[k,j]=sign(y[j]<=t[k])}
z[p,k]=sum(w[k,])/n}}
z=as.data.frame(z)
r=c(z$V1,z$V2,z$V3,z$V4,z$V5,z$V6,z$V7,z$V8,z$V9)
e=c(rep(0.0,100),rep(0.67,100),rep(0.84,100),rep(1.28,100),rep(1.65,100),rep(2.32,100),rep(2.58,100),rep
q=data.frame(T=rep(0,100),X=0)
for(s in 1:900)
{q[s,2]=r[s]}
for(s in 1:900)
{q[s,1]=e[s]}

for(a in 1:100)
 { q[a,2]=q[a,2]-x[1]
```

```
 q[a+100,2]=q[a+100,2]-x[2]
 q[a+200,2]=q[a+200,2]-x[3]
 q[a+300,2]=q[a+300,2]-x[4]
 q[a+400,2]=q[a+400,2]-x[5]
 q[a+500,2]=q[a+500,2]-x[6]
 q[a+600,2]=q[a+600,2]-x[7]
 q[a+700,2]=q[a+700,2]-x[8]
 q[a+800,2]=q[a+800,2]-x[9]}
library(ggplot2)
ggplot(q,aes(T,X,group=T)) + geom_boxplot()
```



when n=10^4,the box plots of the 100 approximation errors at each t

```
x=pnorm(  c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72), mean = 0, sd = 1)
t=c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72)

n=10^4
z=matrix(0,100,9)
w=matrix(0,9,n)

for(p in 1:100)
{ y=c(rnorm(n,mean=0,sd=1))
  for(k in 1:9)
  {
   for(j in 1:n)
  {w[k,j]=sign(y[j]<=t[k])}
```

```
z[p,k]=sum(w[k,])/n}}
z=as.data.frame(z)
r=c(z$V1,z$V2,z$V3,z$V4,z$V5,z$V6,z$V7,z$V8,z$V9)
e=c(rep(0.0,100),rep(0.67,100),rep(0.84,100),rep(1.28,100),rep(1.65,100),rep(2.32,100),rep(2.58,100),re
q=data.frame(T=rep(0,100),X=0)
for(s in 1:900)
{q[s,2]=r[s]}
for(s in 1:900)
{q[s,1]=e[s]}

for(a in 1:100)
 { q[a,2]=q[a,2]-x[1]
 q[a+100,2]=q[a+100,2]-x[2]
 q[a+200,2]=q[a+200,2]-x[3]
 q[a+300,2]=q[a+300,2]-x[4]
 q[a+400,2]=q[a+400,2]-x[5]
 q[a+500,2]=q[a+500,2]-x[6]
 q[a+600,2]=q[a+600,2]-x[7]
 q[a+700,2]=q[a+700,2]-x[8]
 q[a+800,2]=q[a+800,2]-x[9]}
library(ggplot2)
ggplot(q,aes(T,X,group=T)) + geom_boxplot()
```